1969

Verification of model of molten glass flow in a forehearth.

Pekcelen, Sina

Monterey, California. U.S. Naval Postgraduate School

VERIFICATION OF MODEL OF MOLTEN GLASS FLOW IN A FOREHEARTH

by

Sina Pekcelen

United States Naval Postgraduate School

THESIS

VERIFICATION OF MODEL OF MOLTEN GLASS FLOW IN A FOREHEARTH

by

Sina Pekçelen

December 1969

Verification of Model of Molten Glass Flow

in a Forehearth

by

Sina Pekçelen
Lieutenant (junior grade), Turkish Navy
B.S., Naval Postgraduate School, 1969

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN CHEMISTRY

from the

NAVAL POSTGRADUATE SCHOOL
December 1969

ABSTRACT

In this work, variations on velocity profiles in a flowing mass of molten glass in a forehearth are investigated. Formerly used parabolic velocity profiles are replaced with analytical solution of open channel flow equation, based on the available data on mass flow of molten glass through the channel in unit time.

Concerning the viscosity effects; temperature dependence of viscosity is built in the model. However, it is assumed that the depth of the channel does not affect the viscosity gradients.

To solve the system of non-linear differential equations i.e., heat equation and flow equation; the analytical solution of the latter at the nodes is used for the numeric solution of the former iteratively, until the convergence is obtained. Predicted temperatures are compared to the available data from an actual operating forehearth, and against the results predicted by the previous model using simplying assumptions to prove its validity.

TABLE OF CONTENTS

LIST OF ILLUSTRATIONS

TABLE OF SYMBOLS

A	Radiating area
C	Specific heat
d	Actual channel height
F	Radiating enclosure view factor
h	Channel height - general or convective heat transfer coefficient
k_c	Thermal conductivity - ceramic material
k_{eff}	Effective thermal conductivity
k_{rad}	Radiation thermal conductivity
k_{true}	True thermal conductivity
n	Refractive index
P	Pressure
q	Radiant heat transfer rate
R	Aspect ratio
T	Temperature
T_c	Temperature - ceramic material
T_s	Temperature of radiating surface
V or u	Velocity
$\bar{V}$ or $\bar{u}$	Average velocity
w	Channel width
W_v	Radiant volume emmissive power
γ	Radiant absorption coefficient
ε	Value for convergence criteria
ε_G	Emissivity of radiating glass surface

ϵ_s	Emissivity of radiating ceramic surface
θ	Time
λ	Radiation wave length
ρ	Density
σ	Boltzmann constant
μ	Viscosity

ACKNOWLEDGEMENT

The author wishes to express his appreciation for guidance and encouragement given him by Professor John Duffin of the Naval Postgraduate School.

I. INTRODUCTION

To improve the production of glass containers, and to transform the art of glass making into a science, the glass industry embarked an extensive forehearth measurement program in an attempt to resolve the complex phenomena of glass conditioning and cooling, while molten glass is taken out of the furnace and transferred to the forming machine.

The objective of the program was the collection of quantitative data to achieve greater understanding of forehearth and to verify the mathematical model of molten glass flow in forehearth developed by Duffin [Ref. 1].

A knowledge of the temperature distribution is needed for better understanding of the control process in glass plants. The above model proved that sophisticated models can give satisfactory results describing the forehearth behavior in a temperature sense. However they need a big computer and require considerable time to run.

In this work, validity of the simplifying assumptions in [Ref. 1] are investigated and ways to relieve them are sought. Going to a more complicated model has academic interest to show that what was developed was a useful and fairly accurate model.

II. NATURE OF THE PROBLEM

Forehearth is that section of the process where molten glass is transferred from furnace to the forming machine. During the flow, glass is conditioned in a temperature sense. A predetermined temperature is desired at the forming machine end, which assures minimum off-grade material after forming process. If undesirable temperature gradients exist at the input of forming machine, recycling of off-grade glassware is necessary, decreasing the overall system efficiency.

What is needed is controlled conditioning throughout the forehearth while molten glass is flowing. This is mainly a heat transfer problem on a flowing mass of molten glass.

To describe the physical phenomenon taking place in a typical forehearth used in glass container manufacture, see Figure (1). Glass flows from left to right in a open channel with dimensions approximately 26 inches wide, 6 inches deep and 18 feet long.

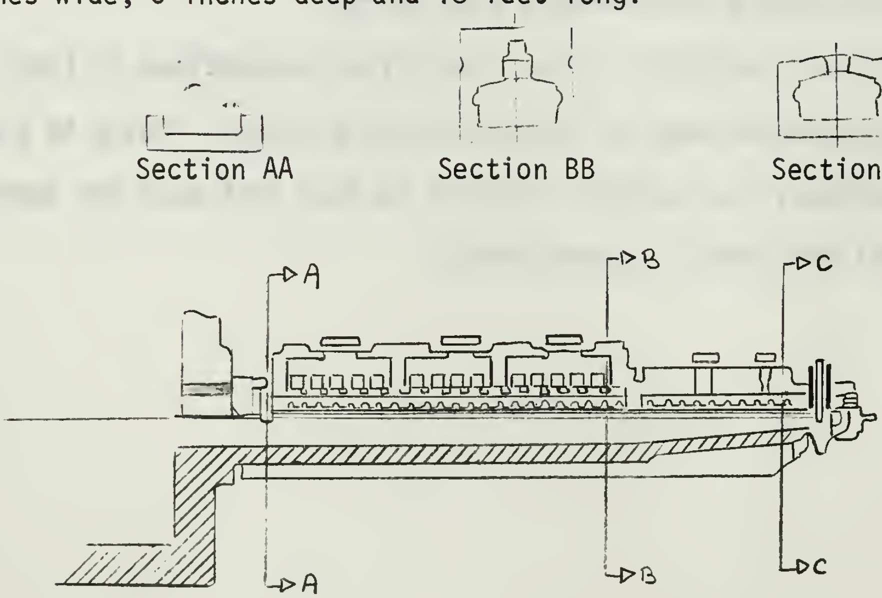


Figure 1. Typical type K forehearth

Section AA is the point of entry from furnace area with average temperature being 2400 degrees F. at this point. Area from AA to BB is called cooling zone. Area from BB to CC is known as conditioning zone. This is the place where an attempt is made to condition the glass so that the delivery temperature would be constant in plane CC. After CC a glass gob is formed. A gob is a discrete mass of molten glass created by intermittenly shearing a stream of molten glass coming from an orifice. The gob formed falls into a guide chute, and is conducted to a forming machine mold.

Temperature control is achieved by adjusting burner flame-levels and the amount of cooling wind in the cooling zone (section AA to BB); also with burner flame levels in the conditioning zone.

Burners are mounted on manifolds and are located every 4.5 inches on both sides of the forehearth. They produce a blanket of heat over glass surface. Hot gases sweep over the ceramic radiating surface and heat exchange takes place between gases and surface which, in turn exchanges heat with molten glass. Cooling is achieved by introducing cooling air through inlet pipes. The wind exchanges heat with glass and the ceramic surface and goes out of the system through roof ports. Details of the forehearth cross section is shown in Figure (2).

Heat exchange at the ceramic surface takes place by convection and radiation mechanisms. Heat is also conducted through the ceramic material in the sides and bottom of the forehearth. Possible disturbances can affect the system through these walls.

Above discussion indicates a complex problem of heat exchange in forehearths, with boundaries as described.

To describe temperature behavior within the glass, it is necessary to consider the mechanisms by which heat is transferred.

FOREHEARTH TEST SECTION INSTRUMENTATION
END VIEW

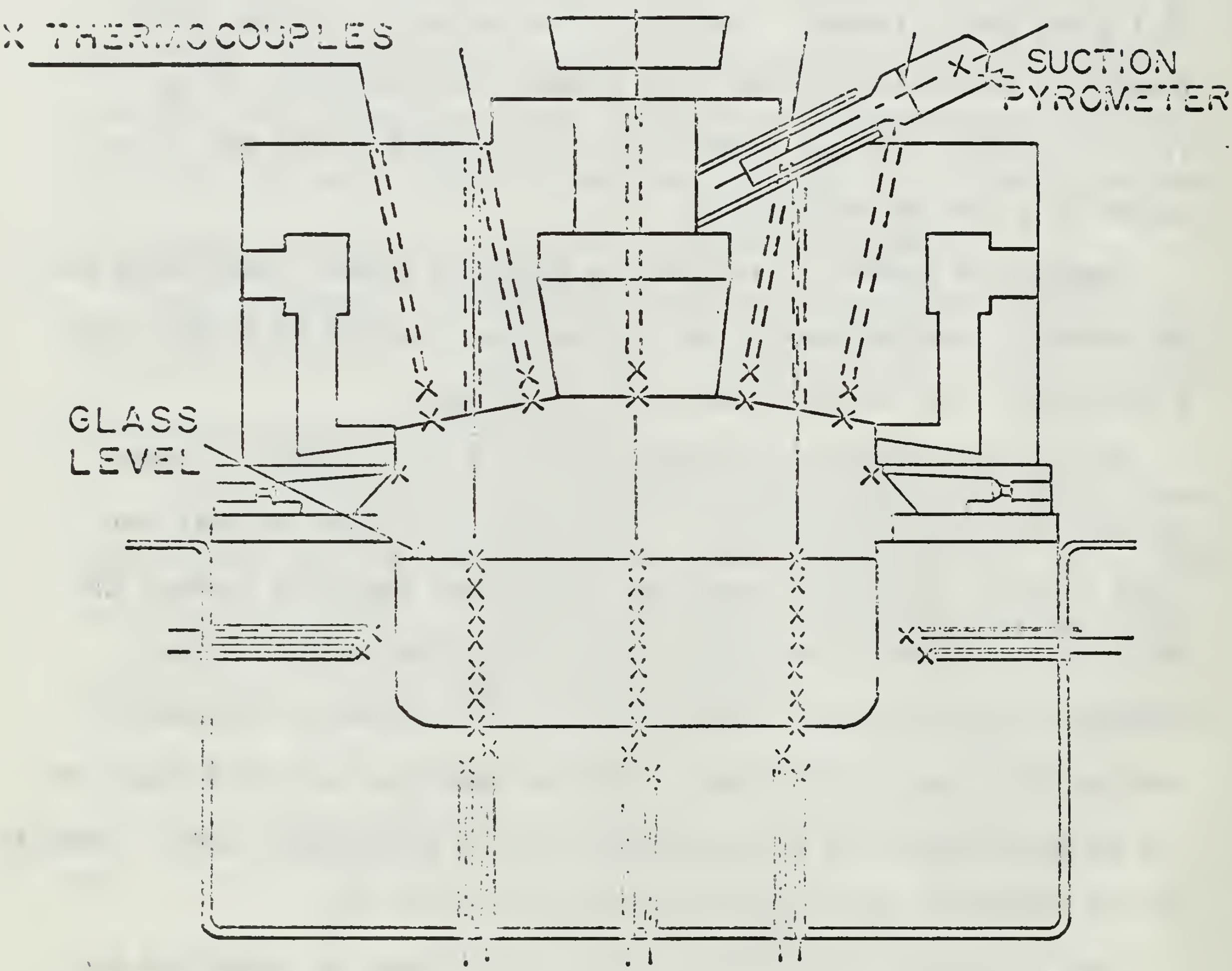


FIGURE 2

It had been shown that emission and absorption of radiation is a bulk phenomena. The interaction of simultaneous emission and absorption throughout the volume leads to radiative heat transfer between adjacent layers of material. When this radiative heat transfer is combined with true conductivity, mathematical description of this internal radiative mechanism becomes rather involved.

Glass is, then, a diathermanous material whose diathermancy is determined by

a) Wave length-absorption coefficient relation and
b) Spectral composition of radiation involved.

This dependency is usually shown in plots of absorption coefficient γ versus wave length λ, with temperature as a parameter. Any detailed analysis of the internal radiation should recognize this varying diathermancy. Integration can be replaced by summation using average values of γ for various ranges of λ, with the average value in a given λ range being a function of temperature. Because of its transparency, glass layers beneath the surface will exchange heat with the surroundings. The depth to which this exchange occurs is appreciable varies with the type of the glass being considered. Upper one third of the glass flowing in a forehearth is considered as exchanging energy directly with the surroundings by Duffin [Ref. 1] in his model.

In the glass body, the radiant effects result from bulk phenomena of emission and absorption. Internally emitted radiation is reabsorbed by the glass directly and also as a result of internal reflections. This process leads to the "radiative conductance" of heat through the glass. Internal emission is characterized by "volume emissive power" which is the rate at which a unit volume of glass emits radiation in all directions. For an ideal gray material volume emissive power is given as,

$$W = 4\gamma n^2 \sigma T^4$$

Thus, it is proportional to absorption coefficient, refractive index and Stefan-Boltzman constant. According to Kellet [Ref. 2], in the interior of massive bodies of molten glass, steady-state temperature distributions turn out to be linear. It was also proposed that the effect of radiative conductivity be designated as an "radiation conductivity". For the gray material it is given by

$$k_{RAD} = \frac{16\ n^2 \sigma T^3}{3\gamma}$$

Thus, the effective conductivity within the material is the sum of the thermal conductivity and radiation conductivity, as follows:

$$k = k_{eff} = k_{true} + k_{rad}$$

This is set and held throughout the glass depth. But due to dependence on the absorption coefficient, variations are possible near the glass surface.

A. DERIVATION OF DIFFERENTIAL EQUATIONS

Derivation of the differential equations was accomplished by Duffin [Ref. 1] as follows; using a right handed coordinate system and taking X coordinate as the flow direction, a differential volume element of dimensions dx, dy, dz is set up. Then writing energy balance with dz = 1.

1. Energy Into Element Due To Mass Flow

   Area in X direction = $dy \cdot dz = dy$

   Mass flow = $V_x(1 \cdot dy)\rho$

   Energy flow = $(\rho dy V_x)(C)(T)$

2. Energy Out Of Element Due To Mass Flow

   $$\frac{\partial}{\partial x}(\rho C V_x T dy)dx + \rho C V_x T dy$$

3. Net Flow Energy

= Input - Output =

$$- \frac{\partial}{\partial x} (\rho C V_x T dx) dy$$

4. Energy Into Element Due To Conduction-Radiation

Area = dx·dz = dx

$$\text{Energy} = - k' dx \frac{\partial T}{\partial y}$$

5. Energy Out Of Element Due To Conduction-Radiation

$$\frac{\partial}{\partial y} (- k' dx \frac{\partial T}{\partial y}) dy + [-k' dx \frac{\partial T}{\partial y}]$$

6. Net Conduction-Radiation Energy

$$\frac{\partial}{\partial y} (k' dx \frac{\partial T}{\partial y}) dy$$

7. Net Total Energy Flow

$$\frac{\partial}{\partial y} (k' \frac{\partial T}{\partial y}) - \frac{\partial}{\partial x} (\rho C V_x T) dx dy = (dx dy \cdot \rho) C \cdot \frac{\partial T}{\partial \theta}$$

so, differential equation is

$$\frac{\partial}{\partial y} (k' \frac{\partial T}{\partial y}) - \frac{\partial}{\partial x} (\rho C V_x T) = \rho c \frac{\partial T}{\partial \theta}$$

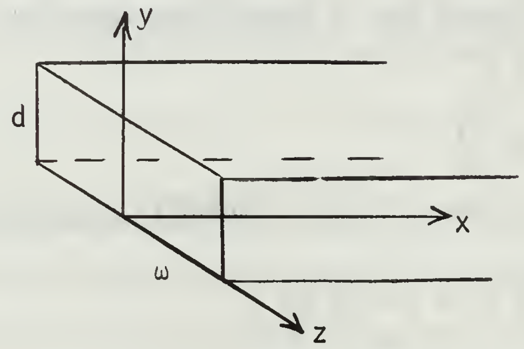
This being one dimensional model, actual case must include the wall effects which is not considered in above derivation. To account for that, a term for Z direction must be included.

$$\frac{\partial}{\partial y} (k' \frac{\partial T}{\partial y}) + \frac{\partial}{\partial z} (k' \frac{\partial T}{\partial z}) - \frac{\partial}{\partial x} (V_x \rho C T) = \rho C \frac{\partial T}{\partial \theta}$$

If $\frac{\partial T}{\partial \theta}$ is equal to zero, steady-state heat flow equation is obtained. In above equation density, specific heat and effective conductivity terms can be treated as temperature independent. Dependence of velocity on temperature will be discussed later.

B. BOUNDARY CONDITIONS

To apply boundary conditions, a fixed coordinate system is placed in channel center as follows



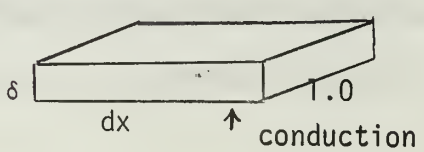
T @	$x = 0$	y and $z = 0$	Temperature in the initial plane
T @	$y = d$	x and $z = 0$	Radiating boundary
T @	$y = 0$	x and $z = 0$	Bottom temperature distribution
t @	$z = w$	x and $y = 0$	Side wall temperature distribution

at $y = 0$ and $z = w$ glass contacts with the ceramic material. The temperature of the glass and ceramic material will be same at these points. Following will also apply

$$\left(\frac{\partial T}{\partial y}\right)_{y=0} = \frac{k_c}{k'}\left(\frac{\partial T_c}{\partial y}\right)$$

$$\left(\frac{\partial T}{\partial z}\right)_{z=w} = \frac{k_c}{k'}\left(\frac{\partial T_c}{\partial z}\right)$$

The boundary at $y = d$ is the radiating boundary. In the model it is assumed that the enclosing surface and the glass surface behave as two infinite opposed parallel planes. If h is the equivalent coefficient of heat transfer for the convection-radiation exchange for the volume element below



Energy Input Area = dx·1

$$-k'dx \left(\frac{\partial T}{\partial y}\right)_{y=\delta} + \sigma\varepsilon_s T_s^{\,4}\, dx$$

Energy Output

$$hdx\,(T - T_s) + \sigma\varepsilon_G T^4\, dx$$

Accumulation

$$-k'dx \left(\frac{\partial T}{\partial y}\right)_{y=\delta} + \sigma\varepsilon_s T_s^{\,4}\, dx - hdx(T - T_s) - \sigma\varepsilon_G T^4 dx = \rho C_p \delta dx\, \frac{\partial T}{\partial \theta}$$

Let $\delta \to o$, so $y \to d$

$$\left(\frac{\partial T}{\partial Y}\right)_{y=d} = \frac{\sigma}{k'}\, F\,(T_s^{\,4} - T^4) - \frac{h}{k'}\,(T - T_s)$$

is the glass to air enclosure boundary condition. F is the view factor which accounts for the geometric arrangement of the two surfaces. With the assumption above following equation applies:

$$F = \frac{1}{1/\varepsilon_s + 1/\varepsilon_G - 1}$$

Where ε_s and ε_G are the emissivities of the enclosing surface and glass, respectively. To go one step ahead, this definition of view factor is replaced with a new one, which also accounts for the re-radiation from the connecting walls. Glass and ceramic surface being gray surfaces connected by nonconducting, reradiating walls, the new shape factor is given by Chapman [Ref. 3] as

$$F = \frac{1}{\dfrac{1}{F_{1-2}} + \dfrac{A_1}{A_2}\left(\dfrac{1}{\varepsilon_s} - 1\right) + \left(\dfrac{1}{\varepsilon_G} - 1\right)}$$

$$F_{1-2} = \frac{A_2 - A_1 F_{1-2}^2}{A_1 + A_2 - 2\, A_1 F_{1-2}}$$

F_{1-2} is the geometric shape factor, A_1 and A_2 are the areas of the radiating surfaces.

C. NUMERICAL METHOD USED TO SOLVE THE HEAT EQUATION

Partial differential equation describing the steady-state temperature $T(x,y,z)$ in a moving rectangular slab of molten glass in a forehearth is

$$\frac{k'}{\rho C_p}\frac{\partial^2 T}{\partial y^2} + \frac{k''}{\rho C_p}\frac{\partial^2 T}{\partial z^2} - \frac{\partial}{\partial x}(V_x T) = 0 \qquad (2.1)$$

Where cartesian coordinates are used, x is the flow direction down the length of the slab from the front of the hearth, y is the direction from the bottom channel toward the top of the slab, and z is the direction from the channel center toward the side. k', k'', C_p are the known constants. The initial temperature distribution at $x = 0$ is assumed to be linear in y and z. It can be found by linear interpolation from the four corner temperatures. Similarly it is assumed that the temperature on the boundaries ($y = 0$), bottom of the channel, and ($z = w$), side of the channel, are also linear. I.e.,

$$T(0,y,z) = \phi_1(y,z) \qquad (2.2)$$

$$T(x,0,z) = \phi_2(x,z) \qquad (2.3)$$

$$T(x,y,w) = \phi_3(x,y) \qquad (2.4)$$

and are all known. The temperature of the glass is symmetric with respect to the center line ($z = 0$); therefore,

$$\left(\frac{\partial T}{\partial z}\right)_{z=0} = 0 \qquad (2.5)$$

Radiative boundary at the surface of the glass ($y = d$) is that of radiative and convective heat transfer. Equation derived previously was,

$$\left(\frac{\partial T}{\partial y}\right)_{y=d} = \frac{\sigma}{k'} F(T_s^{\,4} - T^4) - \frac{h}{k'}(T - T_a)$$

Using linearizing approximation on the nonlinear part, and changing sign on the second term of right hand side,

$$\left(\frac{\partial T}{\partial y}\right)_{y=d} = \frac{4\sigma F}{k'} (T_s + 460)^3 (T_s - T_{y=d}) + \frac{h}{k'} (T_a - T_{y=d}) \quad (2.6)$$

Where σ, F, k', h are known constants. T_a is the air temperature on the glass surface. T_s, the temperature of surroundings is determined by linear interpolation between the temperature at the center of the channel, T_{so} and the known temperature on the boundary at the side (z = w). The center temperature is assumed to decrease linearly from initial temperature T_{s_1} at x = 0 by an amount T_c, at the end of the channel (x = XL); therefore

$$T_{so} = T_{s1} - \frac{X}{XL} T_c$$

With these, the problem is formulated as being the solution of the partial differential equation (1), with initial condition (2.2), and boundary conditions (2.3), (2.4), (2.5), and (2.6).

An unconditionally-stable alternating-direction method [Ref. 4] was applied after passing equation (1) into finite difference form. This is accomplished by first getting the first derivatives with backward differences then substituting these in forward difference equation to get finite difference equation for second space derivatives, in y and z directions. Forward difference form is used for the derivative in x direction.

The first increment step is taken as implicit in y direction, the second implicit in z direction, and so on, x step size being the same in each case. Temperatures at successive planes in x direction are related to each other since the first step involves values at x + Δx and these intermediate values ($T_{i,j}^{*n+1}$)are used in the following step during the numeric calculation. System of equations obtained are,

$$b_{iJ}(T^{*n+1}_{i,J} - T^{n}_{i,J}) = C_1(T^{*n+1}_{i-1,J} - 2T^{*n+1}_{i,J} + T^{*n+1}_{i+1,J}) + C_2(T^{n}_{i,J-1} - 2T^{n}_{i,J} + T^{n}_{i,J+1})$$

$$b_{iJ}(T^{n+1}_{i,J} - T^{*n+1}_{i,J}) = C_1(T^{*n+1}_{i-1,J} - 2T^{*n+1}_{i,J} + T^{*n+1}_{i+1,J}) + C_2(T^{n+1}_{i,J-1} - 2T^{n+1}_{i,J} + T^{n+1}_{i,J+1})$$

With following definitions using increments Δx, Δy, Δz

$$b_{iJ} = V_x(i\Delta x, J\Delta z)$$

$$C_1 = \frac{1/2\Delta x}{\Delta y^2} \frac{k'}{\rho C_p}$$

$$C_2 = \frac{1/2\Delta x}{\Delta z^2} \frac{k''}{\rho C_p}$$

$$T^{n}_{i,J} = T(n\Delta x, i\Delta y, J\Delta z)$$

$$T^{*}_{i,J} = T(n\Delta x, i\Delta y, J\Delta z)$$

The indice i runs from i = 1 to i = I where $i \cdot \Delta y = d$, indice j runs from j = 0 to j = J-1 where $j \cdot \Delta z = w$. The initial temperature gives the values $T^{0}_{i,j} = \phi_1(y,z)$, boundaries give the values $T^{n}_{0,J} = T^{*n}_{0,J} = \phi_2(n\Delta x, J\Delta z)$ when i = I and $T^{n}_{i,J} = T^{*n}_{i,J} = \phi_2(n\Delta x, i\Delta y)$ when J = J-1.

Also, symmetry around the center line of the channel gives rise to $T^{n}_{i,-1} = T^{n}_{i,1}$ and $T^{*n}_{i,-1} = T^{*}_{i,1}$ when j = 0. Radiating boundary results in substitutions for $T^{n}_{i+1,J}$ and $T^{*n}_{i+1,J}$ in terms of $T^{n}_{i,J}$ and $T^{*n}_{i,J}$ when i = I. ϕ_1, ϕ_2 and ϕ_3 are obtained from corner temperatures by linear interpolation.

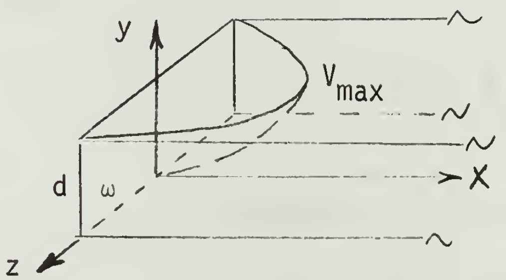
The system of equations is solved starting from $T^{0}_{i,j}$, getting the intermediate solution from the first and using it in the second, making use of Thomas algorithm [Ref. 5] to solve resulting tridiagonal matrices, and proceeding through the channel at every y, z plane separated by

Δx in x direction, until the end of the test section is reached ($x = XL$). For integration purposes x length of the channel is divided into 500 increments ($\Delta x = XL/500$), y coordinate is divided into 30 increments, and z coordinate is divided into 75 increments, making use of the symmetry around the center line.

III. VELOCITY PROFILES

A. PARABOLIC

First considerations of velocity distribution within the flowing mass of molten glass by Duffin [Ref. 1] had involved moving side walls of the channel to infinity which gave velocity in x direction as a simple function of y coordinate. Next step was evaluation of finite velocity step surfaces based on a given mass flow rate of glass in an effort to account for higher velocities in the center channel and lower velocities near the side walls and bottom. None proved satisfactory, so final choice was the use of explicit expressions for parabolic velocity profiles. Physical picture is as follows:



Maximum velocity occurs at y = d and on center line z = 0. Derivation of the general equation of parabola families gives velocity in x direction as a function of y and z, and physical parameters w, d and V_{max} as follows:

$$V_x(y,z) = V_{max}\left[1 - \left(\frac{y}{d}\right)^2 - \left(\frac{z}{w}\right)^2 + \left(\frac{yz}{dw}\right)^2\right] \qquad (3.1)$$

from which 9/4 $V_{ave.} = V_{max.}$ can be obtained. After integration over the channel cross section and dividing by channel area.

B. VELOCITY PROFILES USING OPEN CHANNEL FLOW EQUATION

Above parabolic profile does not account for the viscosity-temperature relationship. Cooper [Ref. 6] pointed out that the effects of moderate

viscosity gradients on velocity are negligible. Temperature in the forehearth changes from 2400°F to 2000°F and across this temperature range viscosity varies by a factor of ten. It has been shown that a viscosity change of 5×10^5 would affect velocity only by factor of three. These facts justify the assumption of constant velocity profiles through the channel length.

To investigate the possible outcome when above assumption is relaxed, parabolic profiles are replaced with the solutions of the differential equation describing laminar flow in open channels. The differential equation to be solved is

$$\frac{\partial}{\partial y}\left(\mu \frac{\partial u}{\partial y}\right) + \frac{\partial}{\partial z}\left(\mu \frac{\partial u}{\partial z}\right) - \frac{dp}{dx} = 0. \qquad (3.2)$$

Where x is the direction of the flow, u is the velocity, and μ is the viscosity. Channel width is taken as w and depth being h. A co-ordinate system is set at the midstream center with boundary conditions:

$$u = 0 \quad @ \quad z = w \qquad (3.3)$$

$$u = 0 \quad @ \quad y = h$$

$$\frac{du}{dy} = 0 \quad @ \quad z = 0$$

$$\frac{dy}{dz} = 0 \quad @ \quad z = 0$$

If viscosity can be considered to be independent of depth and width, above equation simplifies into

$$\frac{\partial^2 u}{\partial y^2} + \frac{\partial^2 u}{\partial z^2} - \frac{1}{\mu}\frac{dp}{dx} = 0 \qquad (3.4)$$

which is solved with its boundary conditions by the use of infinite series. Solution taken from Timoshenko and Goodier [Ref. 7] is

$$u = \frac{16\lambda h^2}{\pi^3} \sum_{n=1,3,5..}^{\infty} \frac{1}{n^3} (-1)^{\frac{n-1}{2}} \left[1 - \frac{\cosh \frac{n\pi z}{2h}}{\cosh \frac{n\pi w}{2h}}\right] \cos \frac{n\pi y}{2h} \qquad (3.5)$$

Integrating over the area of half-section and dividing by the area (h·w) gives the mean velocity $\bar{u}$.

$$\bar{u} = \frac{32\lambda h^2}{\pi^4 w} \sum_{n=1,3,5..}^{\infty} \frac{1}{n^4} \left(w - \frac{2h}{n} \tanh \frac{n\pi w}{2h}\right) \qquad (3.6)$$

Data for the average mass flow rate gives an estimate for $\bar{u}$ value, which in turn can be used to get a value for λ. An experimental value of $\bar{u}$ from GCIRC report [Ref. 8] was 7.77 ft/hour for green glass. This is used in a computer program to evaluate λ taking ten terms of the series (3.6). Result was 0.03649. With that value, velocity at every node of the initial plane is calculated and held constant through the channel, as a first try to change previously used profiles. To give an idea about the difference obtained in velocity profiles, isovels are plotted by 3.0 ft/hour increments in Figure (3b). These changes of main computer program increased the running time considerably. Steady-state results are reported in Appendix (B), under run (A) for $\bar{u}$ = 7.77, run (B) for $\bar{u}$ = 8.54 and run (C) for $\bar{u}$ = 7.00. Purpose of the last two runs was to get an idea about the effect of mass flow rate on the temperature distribution, since the reported value is just a close estimate of the actual mass flow rate in the forehearth. Generally better correlation is observed to actual data in the center line temperatures, but rather poor results observed toward the walls of the channel. Maximum deviation of 15 degrees F. is calculated in these three runs.

C. VELOCITY PROFILES WITH TEMPERATURE DEPENDENCE

To build in the viscosity-temperature dependence, available viscosity data, Appendix (A), is fitted to a functional form as,

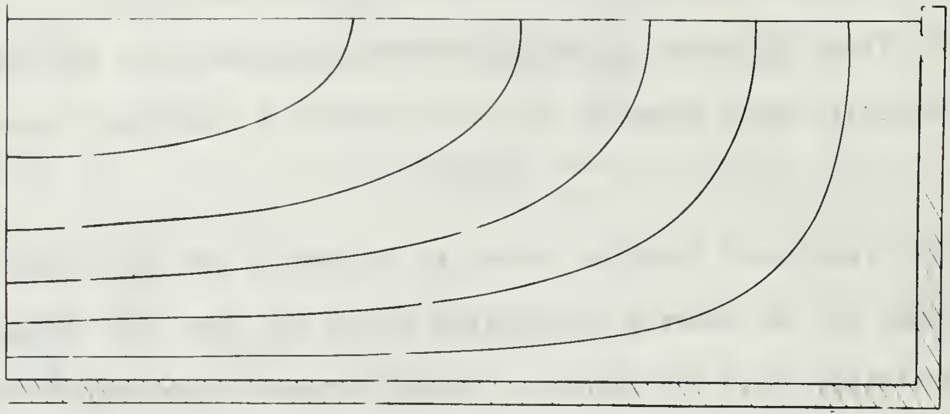
$$\mu = e^{a+bT}$$

This functional form for viscosity is used to get the values of μ at the nodes for the numeric calculation and at the same time employing an estimated value for (dp/dx). Isovels depending on initial temperature distribution are shown in Figure (3c).

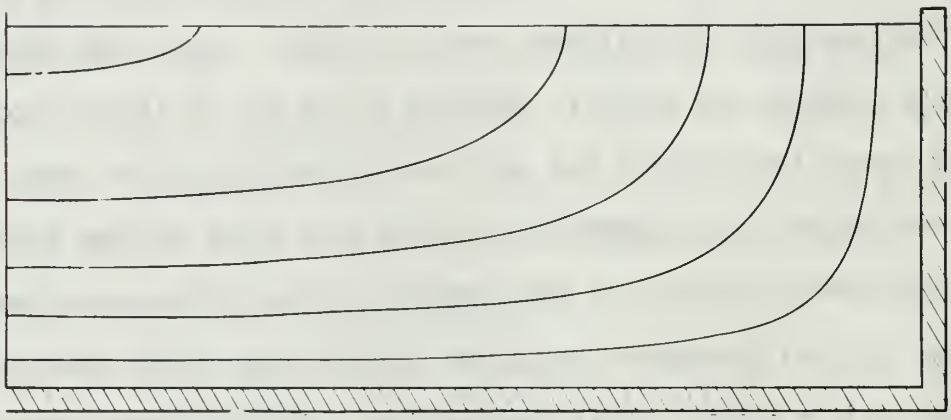
To solve the partial differential equations i.e., heat equation and flow equation, the following route is taken: Start with initial temperature distribution which is obtained by the use of linear interpolation of corner temperatures and calculate velocities at the nodes. Based on these velocities, temperatures in the next plane in flow direction are calculated by the use of the computer program already developed. All the physical parameters are taken as they were in the previous parabolic constant velocity profile computations. Computed temperatures at the nodes are used to recompute the velocity distribution, which is compared with the first velocity array until they agree within the small value ε. Iterative technique is used in the following way: the first velocity array is stored in BA (i,j) array for comparison purposes; after the computation of temperatures in one step down the flow direction, they are used to get the new velocity array BIJ (i,j). If those two do not agree when compared at every node of the plane, a linear combination of the two arrays is taken as

$$BIJ(i,j) = W_a \cdot BA(i,j) + (1 - W_a)BIJ(i,j)$$

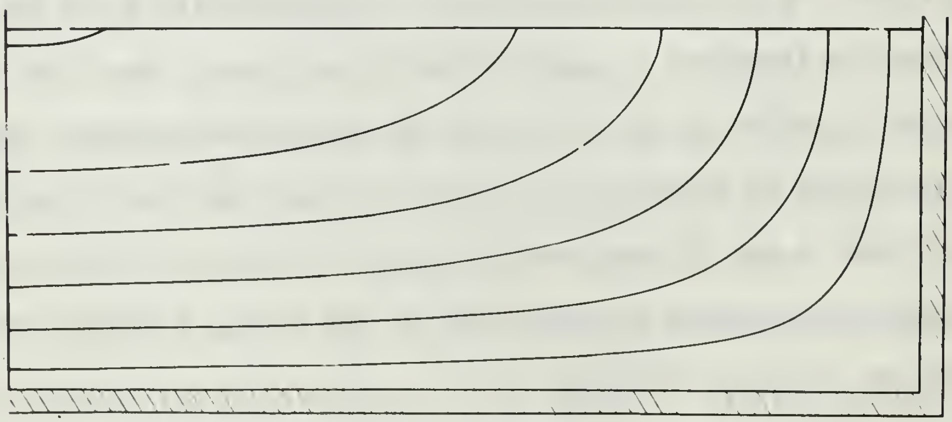
where W_a is a weight factor. The BIJ(i,j) array is stored in BA(i,j)



A. Isovels using Parabolic Equation.



B. Isovels using Open Channel Flow Equation.



C. Isovels with Temperature Dependence

Figure 3. Isovels for Channel Flow

as the new velocity array, and above cycle is repeated until convergence is obtained. Flow diagram is shown in Appendix (D). Numerous runs for the first two planes proved that the above method is appropriate. Value of W_a is varied to have an idea of its influence on the convergence and also on time to run. Final decision was $W_a = 0.25$ since it cuts the running time by half when compared to $W_a = 0.75$.

Due to the inevitable increase in the computer time with the above scheme, instead of updating the velocity array at every Δx increment this is done at every 50 Δx increments. This is updating the velocity array ten times through the channel length. By the trial runs to assign a value to ε, it is found that it takes more than 18 iterations for the convergence with $\varepsilon = 0.001$. Five iterations are needed when $\varepsilon = 0.01$ and 216 seconds to calculate the temperatures at $x = \Delta x$ and $x = 2\Delta x$. Taking these facts into consideration and also considering the reliability of the data, choice of $\varepsilon = 0.1$ is considered appropriate. For run (D) velocity array is corrected with 50 Δx increments, ε value was 0.1, running time 1/4 hour. Results at the nodes where data was taken was comparable to the best run obtained with the constant velocity calculations but not any better. In subsequent runs (D) and (E) the estimate of (dp/dx) is increased by 10% and 15% respectively, giving closer fit to the actual data. The absolute average deviation was around 4.6 degrees F. in all above runs. The largest deviation observed was at y = 5.88 inches and z = 8.63 inches with values around -15 degrees F. This deviation was also present in the results of the previous model. Another variation tried to compute velocity array using 25 Δx increments. Results are reported under run (F). No improvement is observed but running time is increased to twenty minutes.

In run (G), boundary temperature at $x = XL$, $y = d$ and $z = w$ was decreased from 2078°F to 2035°F, and velocity updated with 50 Δx increments. Thus run gave the smallest absolute average deviation 4.2°F between data point temperatures and predicted temperatures. All above runs compute velocity array at the nodes using 10 terms of the defining series equation (3.5). When 5 terms of the series is employed running time decreased more than three minutes. The last run in these lines was run (H) with 5 terms for velocity calculation, ε value of 0.1, and radiating boundary temperature at the channel corner is increased to 2140°F from originally used 2125°F. It took 11.8 minutes to run, maximum deviation was 12.1°F being 1.8°F less than the best result obtained with the previous model. Absolute average deviation from data was 4.22°F. In run (I), previously used view factor is replaced with the new one, accounting for the reradiation from the channel walls. Maximum deviation from the data was 10.6°F, and absolute average deviation was 4.7°F. The predicted temperatures at data points for all above runs are listed in Appendix (B).

VI. PHYSICAL PROPERTY-PARAMETER ESTIMATION

Physical properties used in the equations must be estimated or calculated. Many of these properties and parameters are set at values used in the previous model. Since this work is based on the green glass only, important parameters that were used in original model for that kind of glass are listed below.

Radiation Conductivity	k'	10.0	BTU/HR.FT. F
Density	ρ	147.77	LBS/FT.
Specific Heat	C_p	0.375	BTU/LB. F
Radiant Surface Emissivity	ε_S	0.9	
Glass Surface Emissivity	ε_G	0.9	
Convective Heat Coefficient	h	20.0	BTU/HR.FT. F

In addition to the above, parameters a and b in the functional form $\mu = e^{a+bT}$ are used to get viscosity at the nodes as a function of temperature. Calculated values for different kinds of glass are given below. Plotted viscosity-temperature curve for green glass is presented in Figure (4).

	a	b
Green	9.07	$2.82\ 10^{-3}$
Flint	8.60	$2.60\ 10^{-3}$

To estimate (dp/dx) value; first λ is calculated from the average velocity data based on the mass flow rate, then this value is used together with the average viscosity, since λ is defined as

$$\lambda = \frac{1}{\mu}\,(dp/dx)$$

Calculated value for (dp/dx) is $6.849\ 10^{-4}$.

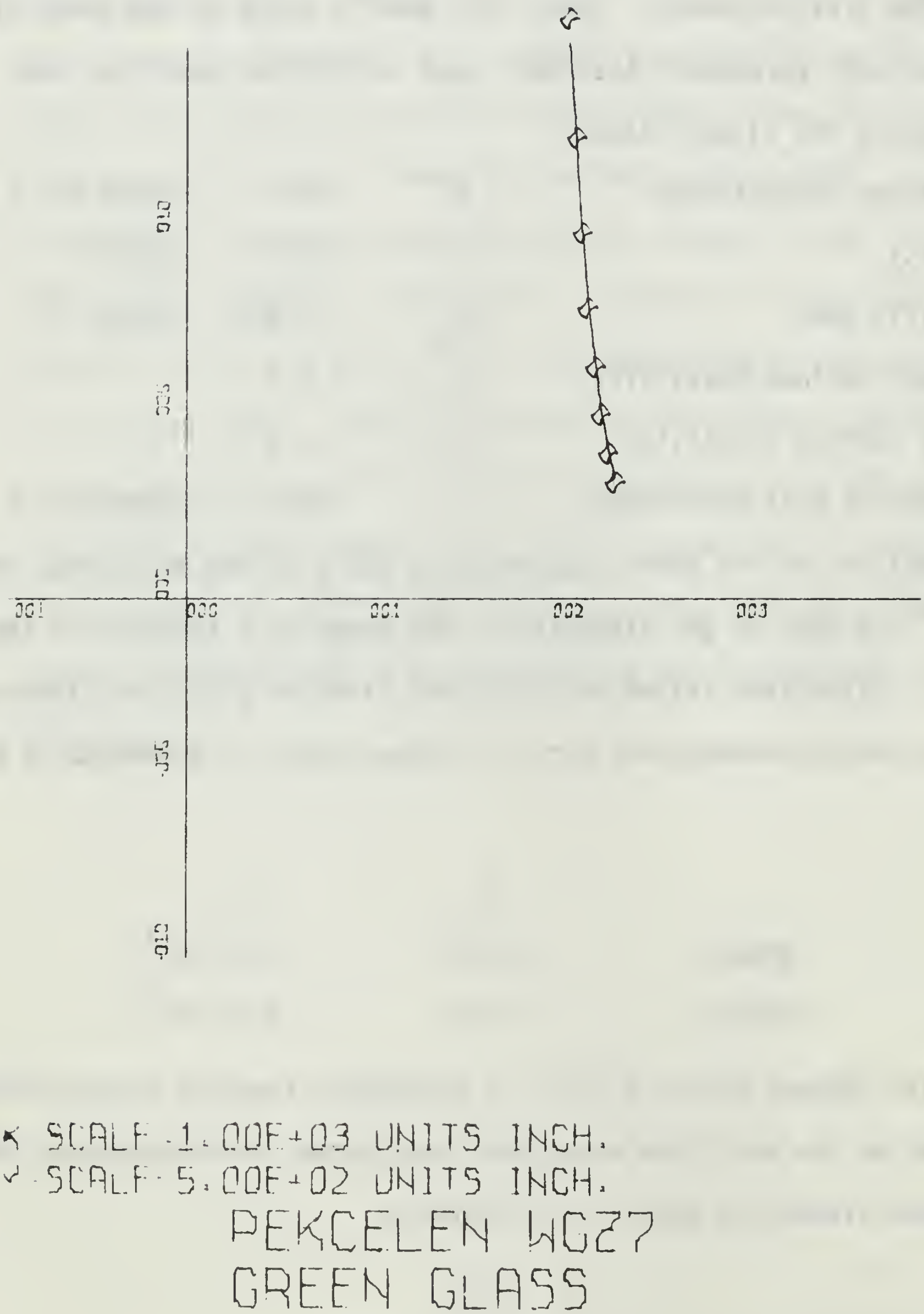


Figure 4. Viscosity-Temp Curve.

In view factor calculations, which account for the reradiation from the channel walls, areas of the radiating surfaces are assumed to be equal. Emissivity values are not changed. Geometric view factor is taken from Chapman [Ref. 3] as 0.56.

V. DISCUSSION

Comparison of the results obtained by the previous and modified models made it clear that the assumption of constant velocity distribution through the channel does not have significant effect on the final results. Predicted temperatures at the nodes were still off by small values. Model generally proved flexible and stable for all variations made of velocities and of boundary conditions. It still does not account for the possible viscosity changes due to pressure changes, since pressure changes with position. However, this is not considered significant since putting in viscosity-temperature dependence proved that initial velocities at the nodes decrease by 20-60% depending on the position. Further work could involve putting in functional forms for density, heat-capacity and thermal-conductivity to build in the temperature dependence of these parameters. If this is done, the model will be further complicated and limit its possible use in a closed loop computer control system. To serve for control purposes, time consumption should be decreased considerably. This will be possible only by the use of some simplifications. If the accuracy in positioning of thermocouples and reliability of the temperature data is considered, the present form of the model gives rather good agreement for the steady-state temperature data. It can possibly be used for design purposes. Requirement for a powerful computer is also a drawback for control purposes. An alternate route might be use of a hybrid computer instead of a digital one to make use of the fast integration abilities of analog computers to decrease the running time.

APPENDIX (A)

GREEN GLASS VISCOSITY DATA

Temperature°F	Viscosity (Centipoises)
2330.4	316.2
2292.9	398.1
2255.5	501.2
2221.1	631.0
2185.7	794.3
2153.1	1000.0
2121.2	1259.0
2089.4	1585.0

APPENDIX (B)

Comparison of data and results predicted with models for green glass

Thermocouple position (in.)
relative to (0,0,0) point

X	Y	Z	Data	Previous Model	Δ	Run(A)	Δ	Run(B)	Δ	Run(C)	Δ	Run(D)	Δ
37.5	5.88	0.0	2127.0	2127.5	+0.5	2123.8	-3.8	2125.9	-1.1	2122.2	-4.8	2123.7	-3.3
		8.63	2095.0	2081.1	-13.9	2083.9	-11.1	2086.6	-8.4	2082.1	-12.9	2079.8	-15.2
37.88	4.88	0.0	2145.0	2146.0	+1.0	2141.7	-3.3	2144.4	-0.6	2138.3	-6.7	2142.5	-2.5
		8.63	2100.0	2096.0	-4.0	2100.7	+0.7	2104.6	+4.6	2097.4	-2.6	2094.8	-5.2
38.25	3.88	0.0	2147.0	2153.0	+6.3	2148.9	+1.9	2152.3	+5.3	2145.0	-2.0	2150.8	+3.8
		8.63	2095.0	2101.0	+6.0	2106.2	+11.2	2110.7	+15.7	2102.1	+7.1	2099.9	+4.9
39.00	1.88	0.0	2145.0	2142.1	-2.9	2139.2	-5.8	2141.6	-3.4	2136.2	-8.8	2140.5	-4.5
		8.63	2077.0	2085.6	+8.6	2088.5	+11.5	2090.0	+13.0	2085.1	+8.1	2082.8	+5.8
39.38	0.88	0.0	2132.0	2129.8	-2.2	2127.8	-4.2	2129.1	-2.9	2126.3	-5.7	2128.1	-3.9
		8.63	2062.0	2069.2	+7.2	2068.4	+6.4	2070.0	+8.0	2066.6	+4.6	2064.9	+2.9
39.75	0.0	0.0	2122.0	2122.0	0.0	2122.0	0.0	2122	0.0	2122.0	0.0	2121.9	-0.1
		8.63	2052.0	2051.7	-0.3	2054.8	+2.8	2047.3	-4.7	2047.3	-4.7	2048.4	-3.6

Thermocouple Position (in.)
relative to (0,0,0) point

X	Y	Z	Data	Run(E)	Δ	Run(F)	Δ	Run(G)	Δ	Run(H)	Δ	Run(I)	Δ
37.50	5.88	0.0	2127.0	2124.9	-2.1	2120.4	-6.6	2126.9	-0.1	2125.3	-1.7	2130.1	+3.1
		8.63	2095.0	2081.0	-14.0	2076.5	-18.5	2079.2	-15.8	2082.9	-12.1	2086.8	-8.2
37.88	4.88	0.0	2145.0	2144.4	-0.6	2138.3	-6.7	2144.5	-0.5	2144.6	-0.4	2148.2	+3.2
		8.63	2100.0	2097.2	-2.8	2090.2	-9.8	2094.0	-6.0	2098.9	-1.1	2101.7	+1.7
38.25	3.88	0.0	2147.0	2152.8	+5.8	2146.2	-0.8	2152.8	+5.8	2153.3	+6.3	2155.3	+8.3
		8.63	2095.0	2102.4	+7.4	2094.7	-0.3	2098.9	+3.9	2103.0	+8.0	2105.6	+10.6
39.00	1.88	0.0	2145.0	2141.5	-3.5	2136.1	-7.9	2141.5	-3.5	2141.9	-3.1	2142.4	-2.6
		8.63	2077.0	2084.6	+7.6	2079.2	+2.2	2082.3	+5.3	2085.2	+7.2	2085.5	+8.5
39.38	0.88	0.0	2132.0	2128.8	-3.2	2126.5	-5.5	2128.8	-3.2	2128.5	-3.5	2129.3	-2.7
		8.63	2062.0	2065.8	+3.8	2063.2	+1.2	2064.5	+2.5	2066.2	+4.2	2066.4	+4.4
39.75	0.0	0.0	2122.0	2122.0	0.0	2121.7	-0.3	2122.0	0.0	2122.0	0.0	2122.0	0.0
		8.63	2052.0	2048.7	-3.3	2048.6	-3.4	2048.7	-3.3	2048.9	-3.1	2048.9	-3.1

APPENDIX (C)

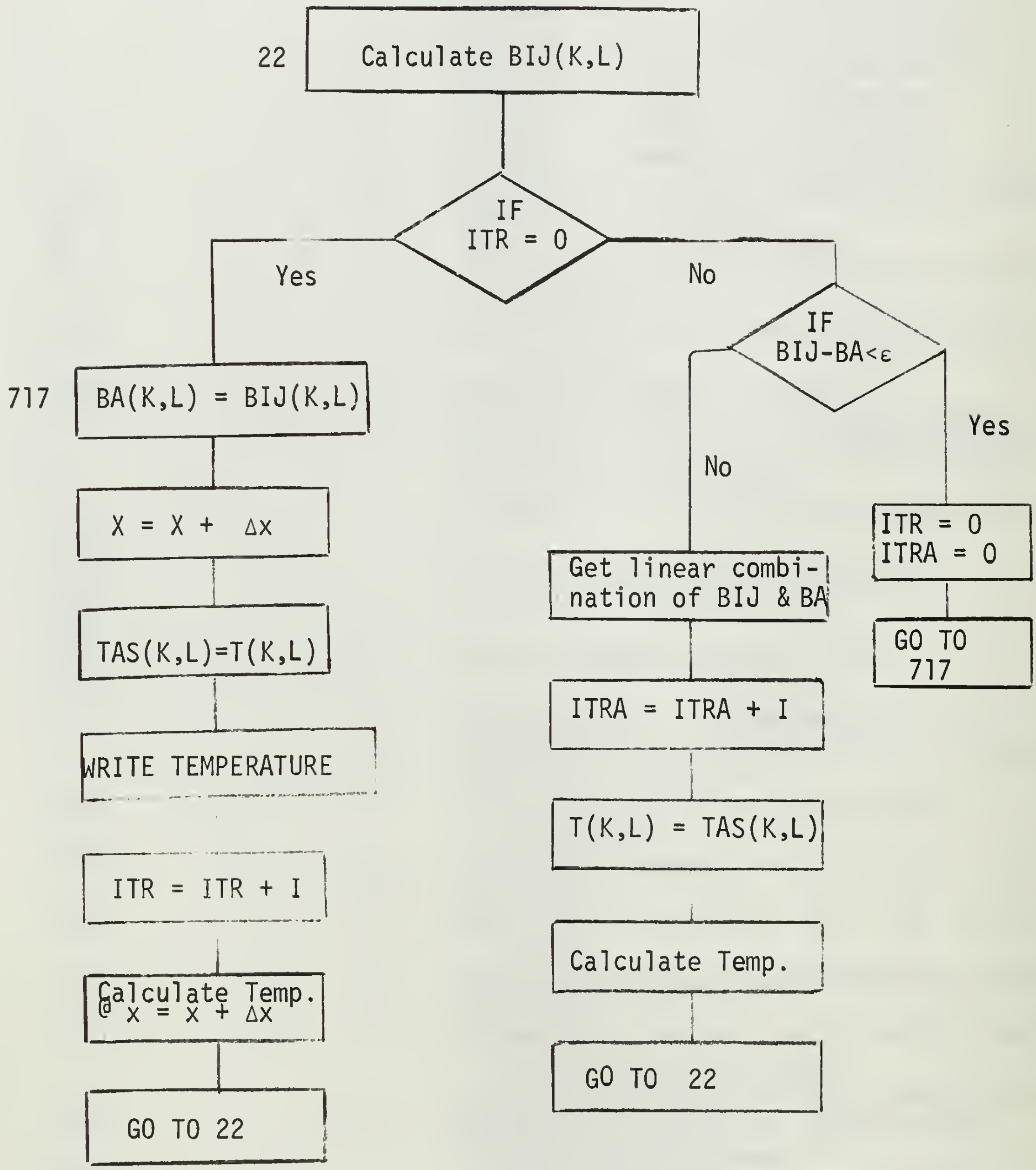
FORTRAN VARIABLE	DEFINITION
N	Maximum value of X direction index n
I	Maximum value of Y direction index i
J	Maximum value of Z direction index j
XL	Length of channel in X direction
YL	Length of channel in Y direction
ZL	Length of channel in Z direction
XEND	Value of X at which to terminate integration
RHO	Density of the glass
CP	Specific heat of the glass
TKP	Glass equivalent thermal conductivity in Y direction
TKPP	Glass equivalent thermal conductivity in Z direction
SIGMA	Stefan-Boltzmann Constant
H	Convective heat-transfer coefficient
ES	Emissivity of ceramic radiating surface
EG	Emissivity of the glass
TSI	Temperature of radiating boundary @ X = 0
TOOO TOOW TODO TODW	Initial plane corner temperatures
TLDW TLOW	Corner temperatures @ x = XL
TA	Temperature of the ambient gas
TC	Temperature of radiating surface along channel centerline
VMAX	Max. glass velocity for parabolic distribution
TODWR	Radiating surface Tem. @ x = 0, y = YL, z = ZL
TLDWR	Radiating surface Tem. @ x = XL, y = YL, z = ZL
IRNT	Counter for steady-state program printout control
NTERM	Number of terms to used in defining series for velocity calculation
ITR	Counter to recalculate the velocity array
ITRA	Counter for number of iterations for velocity convergence
ITRB	Counter used for non-converging velocities at the nodes
NUP	Counter to update the velocity array
WA	Weight factor for linear combination
T(i,j)	2-Dimensional temperature array

BIJ (i,j)	2-Dimensional array for glass velocity (V_x) in X direction at (i,j) points in y,z plane
BA (k,j)	2-Dimensional array for intermediate velocities
TAS (i,j)	2-Dimensional array to store tem. @ the former plane

APPENDIX (D)

FLOW GRAPH TO CALCULATE VELOCITIES

ITR = 0 (Set counters to zero)
ITRA = 0



```
C COMBINED STEADY-UNSTEADY STATE GLASS PROGRAMS.

      DIMENSION T(45,100),BIJ(45,100),DEN1(45,100),DEN2(45,100),
     1 TLAM(45,100),PREM(45,100) ,BA(45,100),TAS(45,100), H(30,75)   ,
     1         JLOT(18),YP(50),TP(50),WD(18),DELS(8)
      DIMENSION TU(30,15,15),S(30,15,15),ALFI(15),DENI(15),ALFJ(15)
      DIMENSION A(30,15),B(30,15),DENJ(15),NOUT(12)
      CCMMON  TU,N,I,J,IPRT,IYP,IZP
    1 FORMAT (12I5)
    2 FORMAT (4E15.6)
    3 FORMAT (1H1,50X5HINPUT///,40X3HN =I6,4H I =I6,4H J =I6)
    4 FORMAT (///,7X2HXL,13X2HYL,13X2HZL,12X4HXEND,12X3HRHO,13X2HCP,
     112X3HTKP,11X4HTKPP/8E15.5)
    5 FORMAT (///,4X5HSIGMA,14X1HH,13X2HES,13X2HEG,12X3HTS1/5E15.5)
    6 FORMAT (///,5X4HTOOO,12X4HTOOW,12X4HTODO,12X4HTODW,12X4HTLOO,
     112X4HTLOW,12X4HTLDW/8E16.5)
    7 FORMAT (///,7X2HTA,14X2HTC,12X4HVMAX,11X5HTODWR,11X5HTLDWR/8E16.5)
    9 FORMAT (///,7X7HJLOT(K)/18I5)
   12 FORMAT(1H0,20X5HIYP =I2,5X5HIZP =I2,5X7HIPRNT =I3,5X6HIPRT =I3)
   13 FORMAT (// ,6X4HDELX,12X4HDELY,12X4HDELZ,14X2HC1,14X2HC2/5E16.5)
   14 FORMAT(1H1,51HTAPE READ HAS REACHED END POINT - START CALCULATION)
  181 FORMAT (1H1,50X11HTEMP AT X =F7.5)
  182 FORMAT (1H0,50X11HTEMP AT X =F7.5)
   19 FORMAT (18F7.1)
  191 FORMAT (17F7.1)
 700   FORMAT(2E15.8)

C READ ARRAY AND SWITCHING(PATH) DATA

      READ(5,1) N,I,J,NTAPE,NBOTH,NTAPEA,NTIMEA
      READ(5,1) NPATHA,NPATHB,NTIME,IPRT,IYP,IZP,IPRNT

C THIS SECTION FOR TRANSIENT RUNS ONLY USING PRERECORDED RESULTS

      GO TO(204,201),NTAPEA
  201 N=N/IPRT
      I=I/IYP
      J=J/IZP
      DO 202 M=1,N
      DO 203 L=1,J
      READ(15,19,END=1081)  (TU(M,K,L),K=1,I)
  203 CONTINUE
  202 CONTINUE
      CALL DYNA1
 1081 WRITE(5,14)
      CALL DYNA1
```

```
C INPUT-OUTPUT SECTION

  204 READ (5,2) XL,YL,ZL,XEND,RHO,CP,TKP,TKPP
      READ (5,2) SIGMA,H,ES,EG,TS1
      READ (5,2) TOOO,TOOW,TODO,TODW,TLOO,TLOW,TLDW
      READ (5,2) TA,TC,VMAX,TODWR,TLDWR
      READ (5,1) (JLOT(K),K=1,18)
       READ(5,1)NTERM
       READ(5,700) PI,FRICA
      WRITE (6,3) N,I,J
      WRITE (6,4) XL,YL,ZL,XEND,RHO,CP,TKP,TKPP
      WRITE (6,5) SIGMA,H,ES,EG,TS1
      WRITE (6,6) TOOO,TOOW,TODO,TODW,TLOO,TLOW,TLDW
      WRITE (6,7) TA,TC,VMAX,TODWR,TLDWR
      WRITE (6,9) (JLOT(K),K=1,18)
      WRITE(6,12) IYP,IZP,IPRNT,IPRT

C INITIALIZATION AND CALCULATION OF VARIOUS SYSTEM CONSTANTS AND PARAMS

      IRNT=IPRNT
      ICNT=IPRT
      IM1=I-1
      JM1=J-1
      DELX=N
      DELX=XL/DELX
      DELY=I
      DELY=YL/DELY
      DELZ=J
      DELZ=ZL/DELZ
      CV1=2.0*DELY/YL*VMAX
      CV2=DELY*DELY/YL/YL*VMAX
      CV3=DELZ*DELZ/ZL/ZL
      C1=TKP*0.5*DELX/DELY/DELY/RHO/CP
      C2=TKPP*0.5*DELX/DELZ/DELZ/RHO/CP
      WRITE (6,13) DELX,DELY,DELZ,C1,C2
      C12=2.0*C1
      C22=2.0*C2
       FRICB=(1.0-FRICA*FRICA)/(2.0*(1.0-FRICA))
       FRIC=1.0/(1.0/FRICB+(1.0/ES-1.0)+(1.0/EG-1.0))
      FH1=-C1*DELY*4.0*SIGMA*FRIC/TKP
      FH3=-C1*DELY*H*TA/TKP
      TH1=-C1*(1.0-H*DELY/TKP)
      A1A=DELX*TC/XL
      A2A=C1*C1
      TS1=TS1+0.5*A1A
```

```
C CALCULATE BOUNDARY CONDITION CONSTANTS AND DO SOME INITIALIZATION
      DELXH=0.5*DELX
      WBI=(TLOW-TOOW)/XL
      WTI=(TLDW-TODW)/XL
      YOLI=(TLOO-TOOO)/XL
      YORI=(TLOW-TOOW)/XL
      WTIR=(TLDWR-TODWR)/XL
      X=-DELX
      M=1
      KOUNT=1
C INITIALIZE T
       NA=0
       ITR=0
       ITRA=0
       ITRB=0
      YN=I
      Z=0.0
      L=1
   20 TTOP=TODO+Z/ZL*(TODW-TODO)
      TBOT=TOOO+Z/ZL*(TOOW-TOOO)
      TINC=(TTOP-TBOT)/YN
      T(1,L)=TBOT+TINC
      DO 21 K=2,I
   21 T(K,L)=T(K-1,L)+TINC
      Z=Z+DELZ
      L=L+1
      IF (L-J) 20,20,22
 CALCULATE VELOCITY DISTRIBUTION DEPENDING ON TEMPERATURE DISTIBUTUON
  22   SUMT=0.0
       NUP=0
      NLIM=NTERM*2-1
       DO  703 K=1,I
       DO 704 L=1,J
       TLAM(K,L)=(135700.0)/(10.0**(9.072-0.00282*T(K,L)))
       PREM(K,L)=TLAM(K,L)*0.129
 704   CONTINUE
  703  CONTINUE
       AY=YL
       DO33 K=1,I
       AY=AY-DELY
       AZ=0.0
```

```
      DO 32 L=1,J
     SUMC=0.0
     DO 702 N=1,NLIM,2
     CUBE=N*N*N
     NSIGN=(-1)**((N-1)/2)
      ARG=(N*PI)/(2.0*YL)
      T1=COSH(ARG*AZ)
      T2=COSH(ARG*ZL)
      T3=COS(ARG*AY)
     T4=1.0-T1/T2
     TERM=NSIGN*T4*T3/CUBE
     SUMC=SUMC+TERM
 702 CONTINUE
      BIJ(K,L)=PREM(K,L)*SUMC
      AZ=AZ+DELZ
  32  CONTINUE
 33   CONTINUE
      IF(ITR.EQ.NA) GOTO 717
      ITRB=0
      DO 730 K=1,I
      DO 731 L=1,J
      DELV=BA(K,L)-BIJ(K,L)
 749  DELV=ABS(DELV)
747   FORMAT(8E16.5)
      IF(DELV-0.05)731,731,791
 791  ITRB=ITRB+1
 731  CONTINUE
 730  CONTINUE
      WRITE(6,1)ITRB
      IF(ITRB.GT.NA)GO TO 714
      WRITE(6,747)((BIJ(K,L),L=1,J),K=1,I)
 1000 ITR=0
      ITRA=0
      GO TO 717
 714  WA=0.25
      ITRA=ITRA+1
      WRITE(6,1)ITRA
      DO 715 K=1,I
      DO716 L=1,J
      T(K,L)=TAS(K,L)
 716  BIJ(K,L)=WA*BA(K,L)+(1.0-WA)*BIJ(K,L)
 715  CONTINUE
 717  DO 718 K=1,I
      DO 719 L=1,J
 719  BA(K,L)=BIJ(K,L)
 718  CONTINUE
```

```
C CALCULATE DEN2

      DO 38 K=1,I
      DO 37 L=1,J
      IF (L-1) 34,34,35
   34 DEN2(K,L)=1.0/(BIJ(K,L)+C22)
      ALF2=C22*DEN2(K,L)
      GO TO 37
   35 DEN2(K,L)=1.0/(BIJ(K,L)+C22-C2*ALF2)
      IF (L-J) 36,37,37
   36 ALF2=C2*DEN2(K,L)
   37 CONTINUE
   38 CONTINUE

C CALCULATE DEN1

      DO 44 L=1,J
      DO 43 K=1,IM1
      IF (K-1) 39,39,40
   39 DEN1(K,L)=1.0/(BIJ(K,L)+C12)
      ALF1=C1*DEN1(K,L)
      GO TO 43
   40 DEN1(K,L)=1.0/(BIJ(K,L)+C12-C1*ALF1)
      ALF1=C1*DEN1(K,L)
   43 CONTINUE
   44 CONTINUE
       IF(ITRA.GT.NA)GO TO 25

C START LOOP

   50 X=X+DELX
       DO 712 K=1,I
       DO 713 L=1,J
 713   TAS(K,L)=T(K,L)
 712   CONTINUE
      GO TO(995,998),NPATHA
  995 IRNT=IRNT+1
      IF(IRNT-IPRNT)  56,65,65

C WRITE T

   65 IRNT=C
  998 IF(X.EQ.0.0) GO TO 999
      GO TO(996,997,999),NPATHB
  997 IF(X.LE.3.12) GO TO 56
      GO TO 999
  996 IF(X.GE.3.12) NPATHA=2
```

```
999 IF(KOUNT-1) 994,994,993
994 WRITE(6,181) X
    KOUNT=2
    GO TO 992
993 WRITE(6,182) X
    KOUNT=1
992 YOLH=TOOO+X*YOLI
    YORH=TOOW+X*YORI
    YOINC=(YORH-YOLH)/ZL*DELZ
    WBO=TOOW+X*WBI
    WTO=TODW+X*WTI
    ZWINC=(WTO-WBO)/YL*DELY
    TEMPW=WBO-ZWINC
    IP1=I+1
    DO 69 K=1,IP1
    DO 67 L=1,18
    J1=JLOT(L)
    IF (J1) 68,68,66
 66 IF (K-1) 46,46,47
 46 D1=J1-1
    WD(L)=YOLH+D1*YOINC
    GO TO 67
 47 WD(L)=T(K-1,J1)
 67 CONTINUE
    L=18
 68 L=L-1
    TEMPW=TEMPW+ZWINC
    WRITE (6,191)(WD(LP),LP=1,L),TEMPW
 69 CONTINUE
 56 GO TO(25,661),NTIME
661 ICNT=ICNT+1
    IF(ICNT-IPRT) 25,23,251
 23 GO TO(242,244),NTAPE
244 DO 243 L=1,J,IZP
    WRITE(15,19)  (T(K,L),K=IYP,I,IYP)
243 CONTINUE
242 GO TO(251,245),NBOTH
245 LL=1
    DO 24  L=1,J,IZP
    KK=1
    DO 241 K=IYP,I,IYP
    TU(M,KK,LL)=T(K,L)
    KK=KK+1
241 CONTINUE
    LL=LL+1
 24 CONTINUE
    M=M+1
251 ICNT=0
```

```
C CHECK X VARIABLE VS. ITS END VALUE

   25 IF (X-XEND) 70,105,105

C CALCULATE F*

   70 DO 76 K=1,I
      DO 75 L=1,J
      LM1=L-1
      IF (LM1) 71,71,72
   71 TM1=T(K,2)
      TNO=T(K,1)
      TP1=TM1
      GO TO 75
   72 T(K,LM1)=C2*(TM1+TP1)+(BIJ(K,LM1)-C22)*TNO
      IF (L-J) 74,73,73
   73 T(K,L)=C2*TNO+(BIJ(K,L)-C22)*TP1
      GO TO 75
   74 TM1=TNO
      TNO=TP1
      TP1=T(K,L+1)
   75 CONTINUE
   76 CONTINUE
      TS1=TS1-A1A
      TH2=(TS1-WTIR*(X+DELXH)-TODWR)*DELZ/ZL
      FH2=TS1+460.0
      PAR1=FH2
      DO 41 L=1,J
      DEN1(I,L)=1.0/(BIJ(I,L)+C12-A2A*DEN1(IM1,L)+TH1-FH1*PAR1**3)
      PAR1=PAR1-TH2
   41 CONTINUE
      PAR1=FH2
      PAR2=TS1
      DO 77 L=1,J
      T(I,L)=T(I,L)-FH3-FH1*PAR2*PAR1**3
      PAR1=PAR1-TH2
      PAR2=PAR2-TH2
   77 CONTINUE
      WBO=TOOW+X*WBI
      WTO=TODW+X*WTI
      ZWINC=(WTO-WBO)/YL
      DZINC=DELY*ZWINC*C2
      D=WBO*C2
      DO 78 K=1,I
      D=D+DZINC
   78 T(K,J)=T(K,J)+D
      YOLH=TOOO+(X+DELXH)*YOLI
```

```
      YORH=TOOW+(X+DELXH)*YORI
      YOINC=(YORH-YOLH)/ZL
      DYINC=DELZ*YOINC*C1
      D=YOLH*C1-DYINC
      DO 79 L=1,J
      D=D+DYINC
   79 T(1,L)=T(1,L)+D
C CALCULATE BETA*
      DO 84 L=1,J
      DO 83 K=1,I
      IF (K-1) 80,80,81
   80 T(K,L)=T(K,L)*DEN1(K,L)
      GO TO 82
   81 T(K,L)=(T(K,L)+C1*BETO)*DEN1(K,L)
   82 BETO=T(K,L)
   83 CONTINUE
   84 CONTINUE
C CALCULATE T*
      DO 86 L=1,J
      JML=J-L+1
      DO 85 K=1,IM1
      IMK=I-K
      T(IMK,JML)=T(IMK,JML)+C1*T(IMK+1,JML)*DEN1(IMK,JML)
   85 CONTINUE
   86 CONTINUE
C CALCULATE F
   87 PAR1=FH2
      PAR2=TS1
      DO 93 L=1,J
      DO 92 K=1,I
      KM1=K-1
      IF (KM1) 88,88,89
   88 TM1=0.0
      TNO=T(1,L)
      TP1=T(2,L)
      GO TO 92
   89 T(KM1,L)=C1*(TM1+TP1)+(BIJ(KM1,L)-C12)*TNO
      IF (K-I) 91,90,90
   90 PAR13=FH1*PAR1**3
      T(K,L)=C1*TNO+(BIJ(K,L)-C12-TH1+PAR13)*TP1-FH3-PAR13*PAR2
      PAR1=PAR1-TH2
      PAR2=PAR2-TH2
```

```
      GO TO 92
   91 TM1=TNO
      TNO=TP1
      TP1=T(K+1,L)
   92 CONTINUE
   93 CONTINUE
      D=YOLH*C1-DYINC
      DO 94 L=1,J
      D=D+DYINC
   94 T(1,L)=T(1,L)+D
      WBN=TOOW+(X+DELX)*WBI
      WTN=TODW+(X+DELX)*WTI
      ZWINC=(WTN-WBN)/YL
      DZINC=DELY*ZWINC*C2
      D=WBN*C2
      DO 95 K=1,I
      D=D+DZINC
   95 T(K,J)=T(K,J)+D

C CALCULATE BETA

      DQ 100 K=1,I
      DO 99 L=1,J
      IF (L-1) 96,96,97
   96 T(K,L)=T(K,L)*DEN2(K,L)
      GO TO 98
   97 T(K,L)=(T(K,L)+C2*BETO)*DEN2(K,L)
   98 BETO=T(K,L)
   99 CONTINUE
  100 CONTINUE

C CALCULATE T

      DO 104 K=1,I
      IMK=I-K+1
      DO 103 L=1,JM1
      JML=J-L
      IF (JML-1) 101,101,102
  101 T(IMK,JML)=T(IMK,JML)+C22*T(IMK,2)*DEN2(IMK,JML)
      GO TO 103
  102 T(IMK,JML)=T(IMK,JML)+C2*T(IMK,JML+1)*DEN2(IMK,JML)
  103 CONTINUE
  104 CONTINUE
       IF(ITRB.GT.NA)GO TO 22
       IF (NUP.EQ.50) GO TO 207
       NUP=NUP+1
       GOTO 50
  207  ITR=ITR+1
```

```
      GO TO 22
C STEADY-STATE PROGRAM DONE.EITHER STOP OR GO ON TO TRANSIENT PROGRAM.
  105 GO TO (1052,1051),NTAPE
 1051 END FILE 15
 1052 GO TO (106,107),NTIMEA
  107 CALL DYNAMO
  106 STOP
      END


C UNSTEADY STATE SECTION OF PROGRAM STARTS HERE.


      SUBROUTINE  DYNAMO
      DIMENSION T(45,100),BIJ(45,100),DEN1(45,100),DEN2(45,100),
     1      JLOT(18),YP(50),TP(50),WD(18)
      DIMENSION TU(30,15,15),S(30,15,15),ALFI(15),DENI(15),ALFJ(15)
      DIMENSION A(30,15),B(30,15),DENJ(15),NOUT(12)
      DIMENSION  TAU(4),CHANGE(4),TAUSB(3),CHGSB(3),DLYSB(3),TSBSS(3)
      DIMENSION  TIME(50),TPLT1(50),TPLT2(50),TPLT3(50),TPLT4(50),TPLT5(
     150),TPLT6(50),TPLT7(50),TPLT8(50),TPLT9(50),TPLT10(50),TPLT11(50),
     2TPLT12(50)
      COMMON  TU,N,I,J,IPRT,IYP,IZP
    1 FORMAT (12I5)
    2 FORMAT (4E15.6)
    4 FORMAT(F10.5,12F10.1)
    5 FORMAT (///,4X5HSIGMA,14X1HH,13X2HES,13X2HEG/4E15.5)
    6 FORMAT (///,5X4HTOOO,12X4HTOOW,12X4HTODO,12X4HTODW/4E16.5)
    7 FORMAT (///,7X2HTA,12X4HVMAX/2E16.5)
    8 FORMAT (///,7X2HXL,14X2HYL,14X2HZL,13X4HTEND,13X3HRHO,14X2HCP,
     113X3HTKP,12X4HTKPP/8E16.5)
    9 FORMAT(//,F19.3,F23.1,F11.1,F15.3)
   10 FORMAT (1H1,50X5HINPUT///,40X3HN =I6,4H I =I6,4H J =I6,5H NT =I6)
   11 FORMAT(1H0,7HPRINT =F7.3)
   12 FORMAT(1H1,12X43HPOSITIONS OF TEMPERATURE READINGS IN INCHES//,4X4
     1HTIME,5X7HX=39.75,3X7HX=39.38,3X7HX=39.00,3X7HX=38.25,3X7HX=37.88,
     23X7HX=37.50,3X7HX=39.75,3X7HX=39.38,3X7HX=39.00,3X7HX=38.25,3X7HX=
     337.88,3X7HX=37.50//,5X2HIN,6X5HY=0.0,5X6HY=0.88,4X6HY=1.88,4X6HY=3
     4.88,4X6HY=4.88,4X6HY=5.88,4X5HY=0.0,5X6HY=0.88,4X6HY=1.88,4X6HY=3.
     588,4X6HY=4.88,4X6HY=5.88//,4X5HHOURS,4X5HZ=0.0,5X5HZ=0.0,5X5HZ=0.0
     6,5X5HZ=0.0,5X5HZ=0.0,5X5HZ=0.0,5X6HZ=8.63,4X6HZ=8.63,4X6HZ=8.63,4X
     76HZ=8.63,4X6HZ=8.63,4X6HZ=8.63,//)
   13 FORMAT (// ,6X4HDELX,12X4HDELY,12X4HDELZ,14X2HC1,14X2HC2/5E16.5)
```

```
   14 FORMAT(///,25X29HRADIATING BOUNDARY CONDITIONS//,35X14HTIME CONSTA
     1NTS//,10X7HTAU(1)=F8.5,7HTAU(2)=F8.5,7HTAU(3)=F8.5,7HTAU(4)=F8.5)
   15 FORMAT(///,25X30HSTEADY-STATE VALUES AT TIME =0//,10X4HTS1=F7.1,
     13HTC=F7.1,6HTODWR=F7.1,6HTLDWR=F7.1)
   16 FORMAT(///,25X43HCHANGES IN STEADY-STATE VALUES AT MAX. TIME//,10X
     110HCHANGE(1)=F6.1,10HCHANGE(2)=F6.1,10HCHANGE(3)=F6.1,10HCHANGE(4)
     2=F6.1)
   17 FORMAT(1H1,25X36HSIDE-WALL-BOTTOM BOUNDARY CONDITIONS//,5X14HTIME
     1CONSTANTS,5X18HSTEADY-STATE VALUE,5X6HCHANGE,5X10HDELAY(HRS))
   18 FORMAT(1H1,10X11HTEMP AT X =F7.5,8H  TIME =F7.5,7H  TS1 =F7.1,
     17H  TS2 =F7.1,6H  TC =F7.1,9H  TODWR =F7.1,9H  TLDWR =F7.1,///)
  183 FORMAT (1H0,50X11HTEMP AT X =F7.5,8H, TIME =F7.5,///)
   19 FORMAT  (18F7.1)
   20 FORMAT(1H1,50X11HTEMP AT X =F7.5,8H  TIME =F7.5,///)

C CALCULATE ARRAY SIZE VARIABLES

      N=N/IPRT
      I=I/IYP
      J=J/IZP

C ENTRY POINT FOR TRANSIENT RUNS ONLY USING PRERECORDED RESULTS
      ENTRY DYNA1

      READ(5,1)  NT
      READ (5,1) (NOUT(L),L=1,12)
      READ (5,2) XL,YL,ZL,TEND,RHO,CP,TKP,TKPP
      READ (5,2) SIGMA,H,ES,EG
      READ (5,2) TOOO,TOOW,TODO,TODW
      READ (5,2) TA,VMAX,PRINT
      READ(5,2)   (TAU(K),K=1,4)
      READ(5,2)   TS1SS,TCSS,DWRSS,TLDWRS
      READ(5,2)   (CHANGE(K),K=1,4)
      READ(5,2) (TAUSB(K),K=1,3)
      READ(5,2) (TSBSS(K),K=1,3)
      READ(5,2) (CHGSB(K),K=1,3)
      READ(5,2) (DLYSB(K),K=1,3)
      WRITE(6,10) N,I,J,NT
      WRITE (6,8) XL,YL,ZL,TEND,RHO,CP,TKP,TKPP
      WRITE (6,5) SIGMA,H,ES,EG
      WRITE (6,6) TOOO,TOOW,TODO,TODW
      WRITE (6,7) TA,VMAX
      WRITE(6,11) PRINT
      WRITE(6,14)  (TAU(K),K=1,4)
      WRITE(6,15)  TS1SS,TCSS,DWRSS,TLDWRS
      WRITE(6,16)  (CHANGE(K),K=1,4)
      WRITE(6,17)
      DO 3 K=1,3
```

```
      WRITE(6,9)  TAUSB(K),TSBSS(K),CHGSB(K),DLYSB(K)
    3 CONTINUE
C INITIALIZATION AND CALCULATION OF VARIOUS SYSTEM CONSTANTS AND PARAMS
      KKK=1
      CNT=0.0
      IM1=I-1
      JM1=J-1
      IP1=I+1
      JP1=J+1
      DELT=NT
      DELT=TEND/DELT
      DELX=N
      DELX=XL/DELX
      DELY=I
      DELY=YL/DELY
      DELZ=J
      DELZ=ZL/DELZ
      CV1=    DELY/YL*VMAX*DELT/DELX
      CV2=DELY*DELY/YL/YL*VMAX*0.5*DELT/DELX
      CV3=DELZ*DELZ/ZL/ZL
      C1=TKP*0.5*DELT/DELY/DELY/RHO/CP
      C2=TKPP*0.5*DELT/DELZ/DELZ/RHO/CP
      WRITE (6,13) DELX,DELY,DELZ,C1,C2
      C12=2.0*C1
      C22=2.0*C2
      C14=2.0*C12
      C24=2.0*C22
      A1=1.0-C14-C24
      FRIC=1.0/(1.0/ES+1.0/EG-1.0)
      C1A=4.0*SIGMA*FRIC*DELY/TKP
      C2A=DELY*H*TA/TKP
      C1B=C1A
      C2B=1.0-DELY*H/TKP
C CALCULATE BIJ
      DO 533 K=1,I
      D=K
      VT1=CV1*D-CV2*D*D
      DO532 L=1,J
      D=L-1
      BIJ(K,L)=VT1*(1.0-CV3*D*D)
  532 CONTINUE
  533 CONTINUE
C CALCULATE DENI AND DENJ
```

```
      DENI(1)=-1.0/(1.0+C12)
      ALFI(1)=C1/(1.0+C12)
      DO523 K=2,IM1
      DENI(K)=1.0/(C1*ALFI(K-1)-1.0-C12)
      ALFI(K)=-C1*DENI(K)
  523 CONTINUE
      ALFI(I)=C1*ALFI(IM1)
      DENJ(1)=-1.0/(1.0+C22)
      ALFJ(1)=C22/(1.0+C22)
      DO524 L=2,JM1
      DENJ(L)=1.0/(C2*ALFJ(L-1)-1.0-C22)
      ALFJ(L)=-C2*DENJ(L)
  524 CONTINUE
      DENJ(J)=1.0/(C2*ALFJ(JM1)-1.0-C22)

C CALCULATE VARIOUS SYSTEM CONSTANTS

      XOTI=(TODW-TOOW)/YL
      XOBI=(TODO-TOOO)/YL
      ZWBI=(TODW-TOOW)/YL
      YOBI=(TOOW-TOOO)/ZL
      Q11=2.0/ZL
      Q12=TOOW-TOOO
      Q13=XOTI-XOBI
      Q14=2.0*TOOO
      Q15=2.0*XOBI
      Q21=1.0/XL
      Q24=TOOO
      Q25=YOBI
      Q31=1.0/XL
      Q34=TOOW
      Q35=ZWBI

C WRITE OUT INITIAL CONDITION DATA

      X=0.0
      Y=0.0
      Z=0.0
      U=0.0
      WRITE (6,20) X,U
      KOUNT=2
      DO529 K=1,IP1
      XOT=TOOW+Y*XOTI
      XOB=TOOO+Y*XOBI
      Y=Y+DELY
      WD(1)=XOB
```

```
      WDI=(XOT-XOB)/ZL*DELZ
      DO528 L=2,J
      WC(L)=WD(L-1)+WDI
  528 CONTINUE
      WRITE (6,19) (WD(L),L=1,J),XOT
  529 CONTINUE
      DO534 M=1,N
      X=X+DELX
      IF(KOUNT-1) 600,600,601
  600 WRITE(6,20) X,U
      KOUNT=2
      GC TO 602
  601 WRITE(6,183) X,U
      KOUNT=1
  602 YOLH=TOOO+X*(TSBSS(1)-TOOO)/XL
      YCRH=TOOW+X*(TSBSS(2)-TOOW)/XL
      WDI=(YORH-YOLH)/ZL*DELZ
      WD(1)=YOLH
      ZWINC=TODW+X*(TSBSS(3)-TODW)/XL
      ZWINC=(ZWINC-YORH)/YL*DELY
      DO530 L=2,J
      WD(L)=WD(L-1)+WDI
  530 CONTINUE
      WRITE (6,19) (WD(L),L=1,J),YORH
      DO531 K=1,I
      YORH=YORH+ZWINC
      WRITE (6,19)(TU(M,K,L),L=1,J),YORH
  531 CONTINUE
  534 CONTINUE
      GO TO 539

C START THE GRAND LOOP

  525 U=U+DELT

C CALCULATE TIME DEPENDENT B.C. TERMS

      TS1=TS1SS+CHANGE(1)*(1.0-EXP(-U/TAU(1)))
      TS2=(TS1SS-TCSS)+CHANGE(2)*(1.0-EXP(-U/TAU(2)))
      TC=TS1-TS2
      TODWR=DWRSS+CHANGE(3)*(1.0-EXP(-U/TAU(3)))
      TLDWR=TLDWRS+CHANGE(4)*(1.0-EXP(-U/TAU(4)))
      C1T=TC/XL
      WTIR=(TLDWR-TODWR)/XL
      TLOO=TSBSS(1)+CHGSB(1)*(1.0-EXP(-(U-DLYSB(1))/TAUSB(1)))
      IF(U.LE.DLYSB(1))   TLOO=TSBSS(1)
      TLOW=TSBSS(2)+CHGSB(2)*(1.0-EXP(-(U-DLYSB(2))/TAUSB(2)))
      IF(U.LE.DLYSB(2))   TLOW=TSBSS(2)
```

```
      TLDW=TSBSS(3)+CHGSB(3)*(1.0-EXP(-(U-DLYSB(3))/TAUSB(3)))
      IF(U.LE.DLYSB(3))  TLDW=TSBSS(3)
      WBI=(TLOW-TOOW)/XL
      WTI=(TLDW-TODW)/XL
      YOLI=(TLOO-TOOO)/XL
      YORI=(TLOW-TOOW)/XL
      ZWTI=(TLDW-TLOW)/YL
      YOTI=(TLOW-TLOO)/ZL
      Q22=TLOO-TOOO
      Q23=YOTI-YOBI
      Q32=TLOW-TOOW
      Q33=ZWTI-ZWBI

C PREPARE RADIATING B.C. TERMS

      X=DELX
      DO536 M=1,N
      Z=0.0
      DO535 L=1,J
      TS=TS1-C1T*X+Z/ZL*(TODWR-TS1+X*(WTIR+C1T))
      TS3=(TS+460.0)**3
      A(M,L)=C1A*TS3*TS+C2A
      B(M,L)=C2B-C1B*TS3
      Z=Z+DELZ
  535 CONTINUE
      X=X+DELX
  536 CONTINUE

C   CALCULATE T*(THIS IS THE RECURSION SECTION OF THE PROGRAM)

      Y=DELY
      DO560 K=1,I
      Z=0.0
      DO559 L=1,J
      X=DELX
      BIJC=BIJ(K,L)
      A1MB=A1-BIJC
      DIV=1.0+BIJC
      DO558 M=1,N
      D=TU(M,K,L)*A1MB
      IF (M-1) 545,545,546
  545 D=D+BIJC*(Q11*Z*(Q12+Q13*Y)+Q14+Q15*Y)
      GO TO547
  546 D=D+BIJC*(S(M-1,K,L)+TU(M-1,K,L))
  547 IF (K-1) 548,548,549
  548 D=D+C12*(TU(M,2,L)+Q21*X*(Q22+Q23*Z)+Q24+Q25*Z)
      GO TO552
  549 IF (K-I) 551,550,550
```

```
  550 D=D+C12*(TU(M,IM1,L)+A(M,L)+TU(M,I,L)*B(M,L))
      GO TO552
  551 D=D+C12*(TU(M,K-1,L)+TU(M,K+1,L))
  552 IF (L-1) 553,553,554
  553 D=D+C24*TU(M,K,2)
      GO TO557
  554 IF (L-J) 556,555,555
  555 D=D+C22*(TU(M,K,JM1)+Q31*X*(Q32+Q33*Y)+Q34+Q35*Y)
      GO TO557
  556 D=D+C22*(TU(M,K,L-1)+TU(M,K,L+1))
  557 S(M,K,L)=D/DIV
      X=X+DELX
  558 CONTINUE
      Z=Z+DELZ
  559 CONTINUE
      Y=Y+DELY
  560 CONTINUE

C     CALCULATE F**

      DO567 L=1,J
      DO566 M=1,N
      DO565 K=1,I
      IF (K-1) 561,561,562
  561 S(M,1,L)=C1*TU(M,2,L)-C12*TU(M,1,L)-S(M,1,L)
      GO TO565
  562 IF (K-I) 564,563,563
  563 S(M,I,L)=C1*TU(M,IM1,L)+(C1*B(M,L)-C12)*TU(M,I,L)-S(M,I,L)
      GO TO565
  564 S(M,K,L)=C1*(TU(M,K-1,L)+TU(M,K+1,L))-C12*TU(M,K,L)-S(M,K,L)
  565 CONTINUE
  566 CONTINUE
  567 CONTINUE

C     CALCULATE BETA**

      DO575 L=1,J
      DO574 M=1,N
      BETO=0.0
      DO573 K=1,I
      IF (K-I) 571,570,570
  570 S(M,I,L)=(S(M,I,L)-C1*BETO)/(ALFI(I)+C1*B(M,L)-1.0-C12)
      GO TO573
  571 S(M,K,L)=(S(M,K,L)-C1*BETO)*DENI(K)
      BETO=S(M,K,L)
  573 CONTINUE
  574 CONTINUE
  575 CONTINUE
```

```
C     CALCULATE T**

         DO578 L=1,J
         DO577 M=1,N
         DO576 K=1,IM1
         IMK=I-K
         S(M,IMK,L)=S(M,IMK,L)+ALFI(IMK)*S(M,IMK+1,L)
     576 CONTINUE
     577 CONTINUE
     578 CONTINUE

C     CALCULATE F

         DO585 K=1,I
         DO584 M=1,N
         DO583 L=1,J
         IF (L-1) 579,579,580
     579 S(M,K,1)=C22*(TU(M,K,2)-TU(M,K,1))-S(M,K,1)
         GO TO583
     580 IF (L-J) 582,581,581
     581 S(M,K,J)=C2*TU(M,K,JM1)-C22*TU(M,K,J)-S(M,K,J)
         GO TO583
     582 S(M,K,L)=C2*(TU(M,K,L-1)+TU(M,K,L+1))-C22*TU(M,K,L)-S(M,K,L)
     583 CONTINUE
     584 CONTINUE
     585 CONTINUE

C     CALCULATE BETA

         DO591 K=1,I
         DO590 M=1,N
         BETO=0.0
         DO589 L=1,J
         TU(M,K,L)=(S(M,K,L)-C2*BETO)*DENJ(L)
         BETO=TU(M,K,L)
     589 CONTINUE
     590 CONTINUE
     591 CONTINUE

C     CALCULATE T

         DO594 K=1,I
         DO593 M=1,N
         DO592 L=1,JM1
         JML=J-L
         TU(M,K,JML)=TU(M,K,JML)+ALFJ(JML)*TU(M,K,JML+1)
     592 CONTINUE
```

```
  593 CONTINUE
  594 CONTINUE
C WRITE TEMPERATURE
      CNT=CNT+1.0
      IF (CNT-PRINT) 537,527,527
  527 CNT=0.0
      DO543 MOUT=1,12
      M=NOUT(MOUT)
      IF (M) 539,539,544
  544 X=M
      X=X*DELX
      IF(KOUNT-1) 603,603,604
  603 WRITE(6,18)  X,U,TS1,TS2,TC,TODWR,TLDWR
      KOUNT=2
      GO TO 605
  604 WRITE(6,183) X,U
      KOUNT=1
  605 YOLH=TOOO+X*YOLI
      YORH=TOOW+X*YORI
      WDI=(YORH-YOLH)/ZL*DELZ
      WD(1)=YOLH
      ZWINC=TODW+X*WTI
      ZWINC=(ZWINC-YORH)/YL*DELY
      DO541 L=2,J
      WD(L)=WD(L-1)+WDI
  541 CONTINUE
      WRITE (6,19) (WD(L),L=1,J),YORH
      DO542 K=1,I
      YORH=YORH+ZWINC
      WRITE (6,19) (TU(M,K,L),L=1,J),YORH
  542 CONTINUE
  543 CONTINUE
  539 TIME(KKK)=U
      TPLT1(KKK)=WD(1)
      TPLT2(KKK)=0.1536*TU(25,3,1)+0.0464*TU(24,3,1)+0.6144*TU(25,2,1)+0
     1.1856*TU(24,2,1)
      TPLT3(KKK)=0.3696*TU(25,5,1)+0.3304*TU(24,5,1)+0.1584*TU(25,4,1)+0
     1.1416*TU(24,4,1)
      TPLT4(KKK)=0.0399*TU(25,10,1)+0.6601*TU(24,10,1)+0.0171*TU(25,9,1)
     1+0.2829*TU(24,9,1)
      TPLT5(KKK)=0.1648*TU(24,13,1)+0.0352*TU(23,13,1)+0.6592*TU(24,12,1
     1)+0.1408*TU(23,12,1)
      TPLT6(KKK)=0.4095*TU(24,15,1)+0.2905*TU(23,15,1)+0.1755*TU(24,14,1
     1)+0.1245*TU(23,14,1)
      TPLT7(KKK)=WD(11)
      TPLT8(KKK)=0.1536*TU(25,3,11)+0.0464*TU(24,3,11)+0.6144*TU(25,2,11
     1)+0.1856*TU(24,2,11)
```

```
      TPLT9(KKK)=0.3696*TU(25,5,11)+0.3304*TU(24,5,11)+0.1584*TU(25,4,11
     1)+0.1416*TU(24,4,11)
      TPLT10(KKK)=0.0399*TU(25,10,11)+0.6601*TU(24,10,11)+0.0171*TU(25,9
     1,11)+0.2829*TU(24,9,11)
      TPLT11(KKK)=0.1648*TU(24,13,11)+0.0352*TU(23,13,11)+0.6592*TU(24,1
     12,11)+0.1408*TU(23,12,11)
      TPLT12(KKK)=0.4095*TU(24,15,11)+0.2905*TU(23,15,11)+0.1755*TU(24,1
     14,11)+0.1245*TU(23,14,11)
      KKK=KKK+1

C CHECK TIME VARIABLE VS. ITS END VALUE AND WRITE INTERPOLATED RESULTS

  537 IF (U-TEND) 525,538,538
  538 KKK=KKK-1
      WRITE(6,12)
      DO  610 I=1,KKK
      WRITE(6,4)   TIME(I),TPLT1(I),TPLT2(I),TPLT3(I),TPLT4(I),TPLT5(I),
     1TPLT6(I),TPLT7(I),TPLT8(I),TPLT9(I),TPLT10(I),TPLT11(I),TPLT12(I)
  610 CONTINUE
      STOP
      END
```

TYPICAL DATA SET FOR STEADY-STATE CALCULATION

```
 500   30   75    1    1    1    1
   2    2    1                  20
      3.3125     0.5             1.083              3.313
        147.77          0.375            10.0            10.0
  0.171300E-08                            0.9             0.9
        2175.0
     2180.0        2167.0          2195.0        2195.0
     2122.0        2010.0          2078.0
                     90.0           17.48         2125.0
     2020.
   1    6   11   16   21   26   31   36   41   46   51   56
  61   66   71   75
   5
 3.14159265     0.56
```

LIST OF REFERENCES

1. International Business Machine Corp., System Development Div., Report 02-472-1, Glass Container Process; Forehearth Simulation by J. Duffin and K. Johnson, 19 July 1965.

2. Kellet, B. S., Steady Flow of Heat Transfer Through Hot Glass, Journal of the Optical Society of America, V. 42, p. 339-343, 21 Nov. 1951.

3. Chapman, A. J., Heat Transfer, p. 334-335, The McMillan Company, 1960.

4. Douglas, Jim, Jr. and D. W. Peaceman, Numerical Solution of Two-Dimensional Heat-Flow Problems, A.I.Ch.E. Jour. V.I., p. 505-507, Dec. 1955.

5. Douglas, Jim, Jr., Advances in Computers, Vol. 2, p. 37-38, Academic Press, 1961.

6. Cooper, A. R., Jr., Effect of Aspect Ratio and Viscosity Gradients on Flow Through Open Channels, Jour. Am. Ceramic Soc., Vol. 43, No. 2, p. 97-104, 19 August 1959.

7. Timoshenko, Stephen and J. N. Goodier, Theory of Elasticity, 2nd ed., p. 275, McGraw-Hill, 1951.

8. Glass Container Industry Research Corporation, Preliminary Study of the Mathematical Model for the Control of Forehearth by T. H. Finger, (Chart 529), 1966.

INITIAL DISTRIBUTION LIST

		No. Copies
1.	Defense Documentation Center Cameron Station Alexandria, Virginia 22314	20
2.	Library, Code 0212 Naval Postgraduate School Monterey, California 93940	2
3.	Professor J. H. Duffin Department of Material Science & Chemistry Naval Postgraduate School Monterey, California 93940	1
4.	LTJG Sina Pekcelen Cevizlik Niyazi Bey Sok No. 24 Bakirkoy-Istanbul Turkey	5
5.	Department of Material Science & Chemistry Naval Postgraduate School Monterey, California 93940	3
6.	Commander,Naval Ordnance Systems Com. Hqs. Department of the Navy, Washington D.C. 20360	1

UNCLASSIFIED
Security Classification

DOCUMENT CONTROL DATA - R & D

(Security classification of title, body of abstract and indexing annotation must be entered when the overall report is classified)

1. ORIGINATING ACTIVITY *(Corporate author)* Naval Postgraduate School Monterey, California 93940	2a. REPORT SECURITY CLASSIFICATION UNCLASSIFIED 2b. GROUP

3. REPORT TITLE

Verification of Model of Molten Glass Flow in a Forehearth

4. DESCRIPTIVE NOTES *(Type of report and inclusive dates)*

Master's Thesis, December 1969

5. AUTHOR(S) *(First name, middle initial, last name)*

Sina Pek..elen

6. REPORT DATE	7a. TOTAL NO. OF PAGES	7b. NO. OF REFS
December 1969	60	8

8a. CONTRACT OR GRANT NO.	9a. ORIGINATOR'S REPORT NUMBER(S)
b. PROJECT NO.	
c.	9b. OTHER REPORT NO(S) *(Any other numbers that may be assigned this report)*
d.	

10. DISTRIBUTION STATEMENT

This document has been approved for public release and sale; its distribution is unlimited.

11. SUPPLEMENTARY NOTES	12. SPONSORING MILITARY ACTIVITY
	Naval Postgraduate School Monterey, California 93940

13. ABSTRACT

In this work, variations on velocity profiles in a flowing mass of molten glass in a forehearth are investigated. Formerly used parabolic velocity profiles are replaced with analytical solution of open channel flow equation, based on the available data on mass flow of molten glass through the channel in unit time.

Concerning the viscosity effects; temperature dependence of viscosity is built in the model. However, it is assumed that the depth of the channel does not affect the viscosity gradients.

To solve the system of non-linear differential equations i.e., heat equation and flow equation; the analytical solution of the latter at the nodes is used for the numeric solution of the former iteratively, until the convergence is obtained. Predicted temperatures are compared to the available data from an actual operating forehearth, and against the results predicted by the previous model using simplying assumptions to prove its validity.

14. KEY WORDS	LINK A		LINK B		LINK C	
	ROLE	WT	ROLE	WT	ROLE	WT
Forehearth Simulation						
Molten Glass Flow						
Math. Model - Numerical Analysis						
Coupled Partial Differential Eqs. (Non-Linear)						

DD FORM 1 NOV 65 1473 (BACK)
S/N 0101-807-6821